Department of Education
Student Financial Assistance

Graphical User Interface

This deliverable is a description of the elements of the graphical user interface (GUI), such as GUI dialog flows, windows, and widgets. Use this deliverable to describe the elements of a GUI. Also use this for prototyping the graphical interface.

When implementing packaged software, this composite deliverable will be used to document changes to the baseline packaged application. Refer to packaged software documentation for information on the baseline packaged application.

I. IPT Name:		
II. Deliverable Name: CAR Diagram		**Date Completed:**
III. Contact Information		
	Name	Channel Unit
IPT Sponsor		
Channel Task Manager		
CIO Task Manager		
Contractor Task Manager		
IV. Task Order Number:		

Description

The Graphical User Interface consists of the following deliverables:

- CAR Diagram
- GUI Dialogue Definition
- Widget Definitions
    - Check Box Definition
    - Combo Box Definition
    - Help Text
    - ICON Definition
    - List Box Definition
    - Menu Item Definition
    - Multi-Line Entry Field Definition
    - Pushbutton Definition
    - Radio Button Definition
    - Single-Line Entry Field Definition
    - Slider Definition
- Window Definition

CAR Diagram

This deliverable lists for each widget, the possible user actions on the widget and the description of the corresponding response (and callback, if applicable). In a web-based application, the CAR diagram also lists the effect of the response. Use this deliverable in graphical user interfaces (GUI) and web-based interfaces to identify all the sponsoring organization programs called as a direct result of the user performing an action at the user interface.

Description

[1] Widget: Name of the widget that accesses the functionality.

[2] Widget Type: The type of the widget. Typical values are Window, Menu Item, Push Button, List Box, Entry Field, Combo Box, Check Box, Radio Button, Slider, and Image (for web-based).

[3] GUI Event: Action taken by the user on the widget to access the functionality.

[4] Response: Action that happens in response to the event.

[5] Effect: The result of the response. This is usually the appearance of a page affected by the response. Note that this column is used for web-based interfaces only.

[6] Callback: Module, class, or operation that the response invokes.

[7] Callback Name: Name of the invoked function in the module or the name of the invoked operation in the class.

[1] Widget	[2] Widget Type	[3] Event	[4] Response	[5] Effect	[6] Callback	[7] Callback Name
Home Page Button	Image	click	link to home page	ABC's Home Page Definition		
Order Now Button	Image	click	invoke ORDER script passing the product number as first parameter	ORDER		
Products Banner	Image	click on product Z1	link to product Z1 page	Products Page Definition		
Products Banner	Image	click on product Z2	link to product Z2 page	Products Page Definition		
Products Banner	Image	click on product Z3	link to product Z3 page	Products Page Definition		
Products Banner	Image	click on product Z4	link to product Z4 page	Products Page Definition		

GUI Dialog Definition

This deliverable illustrates the user's interaction with the application when using a graphical interface design. A dialog in the application contains the flow of data and messages required to complete a logical, or business unit of work. Use this deliverable to define the details of the user interaction with the application, depicting the flow of control between the windows within dialogs. These diagrams can be used during the design of the graphical user interface (GUI). They can be broken down into further detail by taking each window identified in the dialog flow and identifying the data it will contain.

When implementing packaged software, this deliverable will be used to document changes to the baseline packaged application. Refer to packaged software documentation for information on the baseline packaged application.

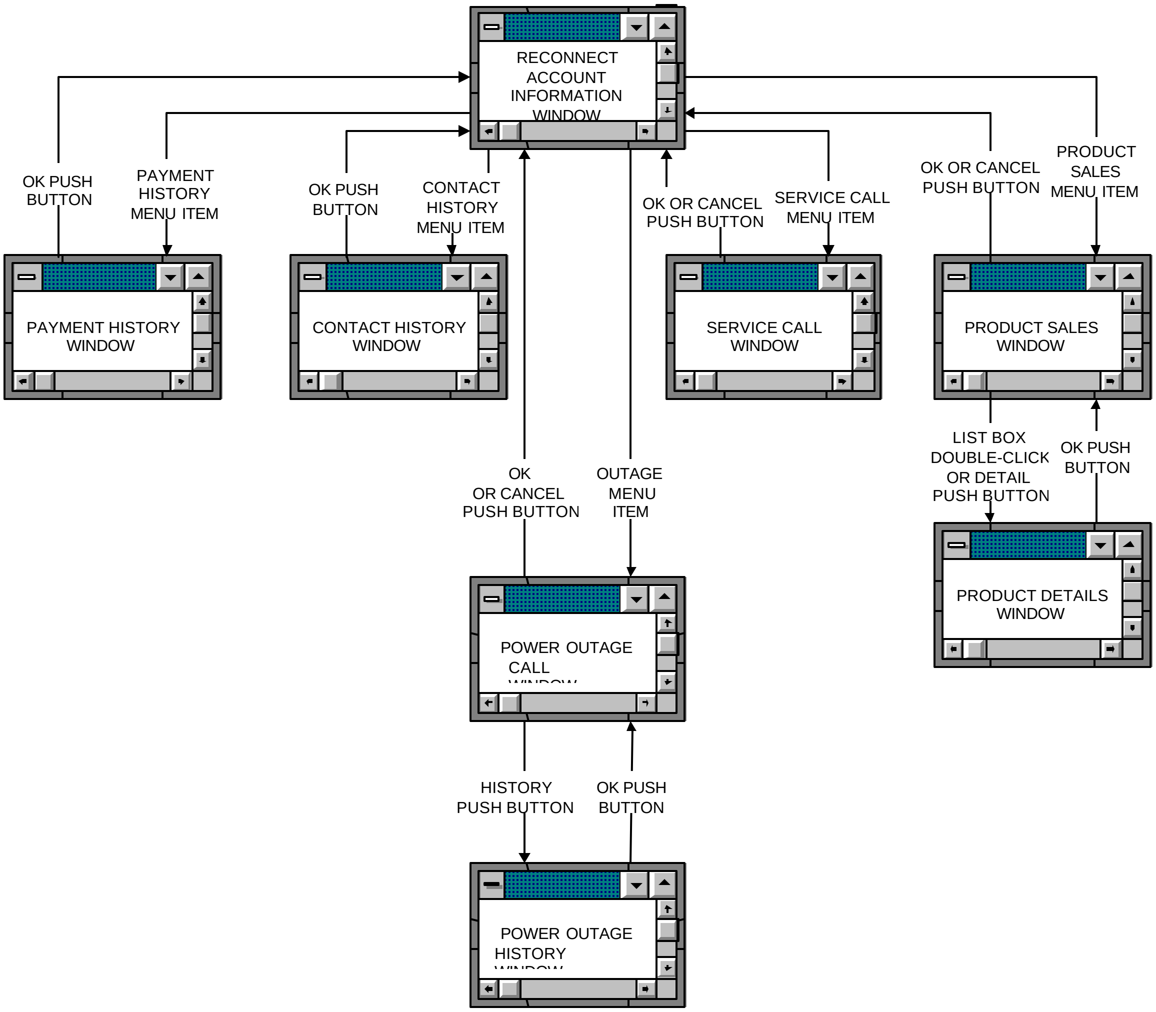
These diagrams can be used at different levels of detail, varying from simple names of windows with arrows, to definitions of functionality, attached to both the windows and the arrows. Because these diagrams can become very complex, they may need to be split across several pages using off-page connectors.

Description

Window: Identifies a window in the dialog and is depicted by labeled rectangles (e.g., "Reconnect Account Information" is a window in the sample).

Arrow: Indicate the window created when the user takes some action. Note, an arrow can connect a window to itself.

Arrow Label: Each arrow is labeled with an identification of the widget and the action that transfers control from one window to another. The OK pushbutton is an example of such a widget.


RECONNECT ACCOUNT INFORMATION WINDOW
OK PUSH BUTTON
PAYMENT HISTORY MENU ITEM
OK PUSH BUTTON
CONTACT HISTORY MENU ITEM
OK OR CANCEL PUSH BUTTON
SERVICE CALL MENU ITEM
OK OR CANCEL PUSH BUTTON
PRODUCT SALES MENU ITEM
PAYMENT HISTORY WINDOW
CONTACT HISTORY WINDOW
SERVICE CALL WINDOW
PRODUCT SALES WINDOW
LIST BOX DOUBLE-CLICK OR DETAIL PUSH BUTTON
OK PUSH BUTTON
PRODUCT DETAILS WINDOW
OK OR CANCEL PUSH BUTTON
OUTAGE MENU ITEM
POWER OUTAGE CALL WINDOW
HISTORY PUSH BUTTON
OK PUSH BUTTON
POWER OUTAGE HISTORY WINDOW

Check Box Definition

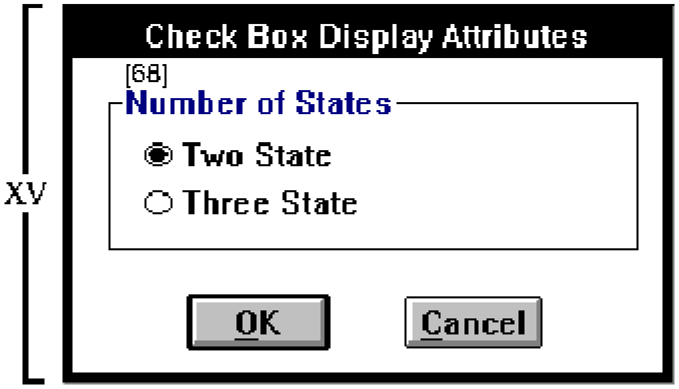
A Check Box Definition describes the possible meanings of an input/output (I/O) data element displayed as a check box widget. Use this deliverable in graphical user interfaces for check box I/O data elements.

Description

In client/server applications, most data elements will appear on one or more windows in the application. Depending on the nature of the data element, it is represented by a particular type of widget, e.g., a single-line entry field, a multi-line entry field, a radio button group, a combo box, a check box, or a slider. The definitions associated with the check box widget type are listed below.

XV: The number of states definable within a check box: As shown in the Check Box Definition sample, a check box data element can be defined as two state (the box is either on or off) or three state (the box can be on, off, or undetermined).

[68] Number of States: An indication of whether a check box will have a two state (yes/no) or three state (yes/no/undetermined) format.



Check Box Display Attributes

[68]
Number of States

● Two State
○ Three State

Combo Box Definition

A Combo Box Definition describes an input/output (I/O) data element displayed as a combo box widget. Use this deliverable in graphical user interfaces for combo box I/O data elements

Description

In client/server applications, most data elements will appear on one or more windows in the application. Depending on the nature of the data element, it is represented by a particular type of widget, e.g., single-line entry field, multi-line entry field, radio button group, combo box, check box, or slider. The definitions associated with the combo box widget type are listed below. Usually, only one of the definitions will apply to any given data element. Since the definitions are mutually exclusive, different examples are illustrated for each one. Note that some of the information also applies to the data elements that appear in traditional screens and reports.

XIV: A description of the display attributes of a combo box: This Combo Box Definition sample provides a partial definition of a combo box to be used in determining the type of view a hotel customer would like from their room's window (pool, ocean, or courtyard). Established here is the type of combo box being defined for this data element (drop-down list) and the box's display attributes. Each field in the combo box will be eight character units wide by 1.5 character units high, and the overall list will be eight character units wide by five character units high.

[51] Field Width: The default width of a single-line entry field, measured in character units. Note that this only refers to the display size of the entry field, and has no bearing on the number of characters the field can actually contain.

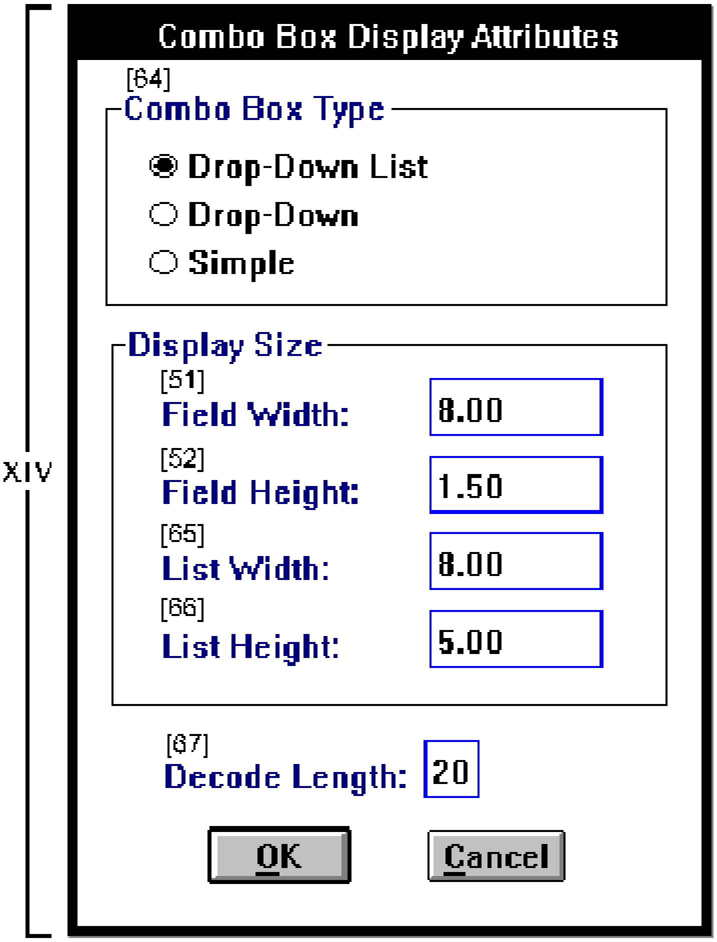
[52] Field Height: The default height of a single-line entry field, measured in character units. Note that this refers only to the display size of the entry field, and has no bearing on the number of characters the field can actually contain.

[64] Combo Box Type: The type of combo box used to represent the data element (drop-down list, drop-down, or simple).

[65] List Width: The width of a combo box selection list as it appears on a screen or window, measured in character units.

[66] List Height: The height of a combo box selection list as it appears on a screen or window, measured in character units.

[67] Decode Length: The maximum number of characters allowable on a single line in the combo box selection list.



Combo Box Display Attributes

[64] Combo Box Type

● Drop-Down List
○ Drop-Down
○ Simple

Display Size

[51] Field Width: 8.00

[52] Field Height: 1.50

[65] List Width: 8.00

[66] List Height: 5.00

[67] Decode Length: 20

Help Text

The Help Text deliverable documents the navigation through the online help that will appear in help sections of the application. The Help Text deliverable also can be used to document features (other than procedures) of the application, such as information about dialog/window features, etc. Use this deliverable to describe how the online help will appear for the business representative. Also use it to describe the integrated help text.

Description

Types of Help

Help can be provided in a number of different ways. Although Help is less interactive than training programs such as tutorials or simulations, it supplies specific information when necessary. Different types of Help include:

- **Fixed Format Help:** Fixed format help is context independent. The same message is displayed, regardless of what the business representative has done. An example would be the explanation of a command. This type of Help involves the creation of text files, are the simplest to implement, and are hidden until the business representative requests help.
- **Context-Sensitive Help:** Context-sensitive Help provides information based on the business representative's current endeavor. For example, if a business representative is trying to create an Excel formula and asks for help, the information presented will be specific to creating formulas. Notice that Context-Sensitive Help is also hidden until the business representative requests help.
- **Prompting Help:** Prompting Help is the same as context-sensitive Help, except that it is automatically generated by the application. For example, the business representative is in DOS and wants to format a disk. **Format b:** is typed and the application prompts that the disk to be formatted be put into the **b** drive. Sophisticated Prompting Help can actually detect and often correct errors.
- **Query-in-Depth Help:** Query-in-Depth Help provides multiple levels of help. Succeeding help requests provide more detail than the former level. Typically, different levels of help can be accessed by setting a parameter in the program or by making successive requests. After each successive level, the business representative is given the option of returning to the application program.
- **Dialogue Help:** Dialogue Help lets the business representatives ask questions in dialogue format. This kind of help depends on some form of natural language capability of the software.
- **Adaptive Help:** Adaptive Help keeps track of how much the business representative knows about the application, and adjusts the level of help information given accordingly. Often, business representatives must demonstrate a mastery of basic operations before other functions are made available to them.

- **Expert Help:** Expert Help analyzes the input, diagnoses the error, infers likely choices, and then presents the business representative with a menu of options of what the business representative really meant to do.

Levels Of Help
Another way to look at help is by levels. Common levels are:

- **Application Overview:** The business representative can obtain an overview of the entire application.
- **Application Level:** The business representative receives help in performing an overall business function.
- **Screen Level:** Describes the entire screen.
- **Field Level:** Describes one field and lists valid entries. The most commonly used form of Help.
- **Table or Code Level:** Defines table fields and how to appropriately enter or retrieve data from a table.
- **User Error Help:** Provides information to the business representative on an error condition encountered, and the action required to correct the error and continue.
- **Notice Board/Message Information:** Advises the business representative at initial sign-on of what changes have been made to the application.
- **Help for Help:** Gives instructions on how to use Help. Often needed with complicated Help applications.
- **Reference Help/Definition Assistance:** Information used to explain or define a term or command, such as a glossary of terms, list of commands, or command descriptions.

Sample

Query in Depth

How Do I...

- o Access Patient Information?
- o Access Subscriber Information?
- o Access Other Carrier Liability (OCL) Information?

How Do I...

Access Patient Information?

To access high-level patient information:

From the Eligibility Verification Request Window:

1. Enter the patient's member number in the Member # field.

2. Enter the provider number of the physician who will be seeing this patient in the Provider # field.

3. Click on the Add pushbutton. This will add high-level patient information to the scrolling list box below the Add pushbutton.

Note: If an error occurs during processing, you will be provided with an error or informational message. If you need further information regarding a message, see the Troubleshooting section of Help.

To obtain a patient's basic contract status:

From the Eligibility Verification Request Window:

1. Enter the patient's member number in the Member # field.

2. Enter the provider number of the physician who will be seeing this patient in the Provider # field.

3. Click on the Add pushbutton. This will add the patient's information to the scrolling list box below the Add pushbutton.

4. Examine the data under the Elig field of the scrolling list box for that patient:
   If the field contains a Y, then the patient has a valid contract.
   If the field contains an N, then the patient does not have a valid contract.

Screen Level / Window Description

EVI01- Eligibility Verification Request

The Eligibility Verification Request window allows you to enter a member number and provider number in order to access contract information. On this window, you can view high-level member information for multiple patients. If the wrong patient information appears in the list box, you can select the member and remove the name from the list. Once the member has been added to the list, you can access additional information by selecting a member from the list and using the buttons at the bottom of the window.

For information on a specific field(s), move the mouse pointer to the area on the Eligibility Verification Request window. When the mouse pointer changes from an arrow to a hand, click the left mouse button.

Fixed Format/Field Description

Call Entry Window

Use this window to briefly document a report as phone/fax, walk-in, mail-in, or worker identified. The report can be a normal request, special request, or actual intake. If the report is a normal or a non-casework-related special request, complete the call on this window.

Tips:
The Intake Toolbar is a row of pushbuttons that appears across the bottom of every Intake window in CAPS. You can use the Toolbar to open Intake windows and record call information in any order.

Keywords or Phrases:

- Call Entry window
- Recording an
- Special request, recording a

Phone Icon

Click on this icon to begin recording a new report. Although the Date and Time fields pre-fill with the current date and time, you may change this information while on the Call Entry window.

Date Field

Use this field to type the date you record a report. It pre-fills with the current date after you click on the Phone Icon. You may change the date while on the Call Entry window. However, if you open another window and then return to the Call Entry window, you cannot change the date.

Phone, Mail/Fax, Walk-in, and Worker (Wkr) ID Radio Buttons

Click on these radio buttons to indicate how you receive a report, either by:

- Phone
- Mail
- Walk-in
- Worker ID

The default radio button is Phone.

First, M, Last, and Suffix (Sfx) Fields

Use the First, M, and Last fields to type the reporter's first, middle, and last name. Also, use the Sfx field to select the suffix of the reporter's name, such as Jr. or III.

User Error Messages Explanations

Warning 4753: Topic...(topic no.) Hidden paragraph

This may or may not be a warning that can be ignored. The true test of whether an actual problem exists will be found within your help file. Check the appearance and functionality of the topic referred to by completing the following steps:

1. Open the .RTF file referred to in your error report (.ERR file).

2. From the **Edit** menu, choose **Go To Topic Number**.

3. Enter the topic number referenced in the error report, and click **OK**.

4. After determining which topic contains a hidden paragraph, open your help file and move to that topic. If the topic referenced appears normal and jumps to and from that topic work correctly, this warning may be safely ignored.

However, if you determine from this test that the appearance and functionality of the help file has been compromised, you must correct the problem, preferably at the document level. To correct the situation, complete the following steps:

1. Locate the problem topic.

2. With paragraph markings visible, select each paragraph mark in turn.

3. With each paragraph marking selected, from the **Format** menu, choose **Font** (or **Character** if working in Word 2).

4. If Hidden is selected, you should deselect that option, and click **OK**.

After reformatting your document, the problem should be resolved.

Icon Definition

An Icon Definition describes the bitmapped, graphical symbol used to represent an object, function, data structure, or user interface element. Use this deliverable when designing a graphical user interface. Use icons to alert business representatives to certain conditions or danger signs (e.g., when used in a message box), or to present a graphical image (e.g., when used in place of a pushbutton or to represent an application). This icon description can serve as input to a code generator.

Icons can be used in windows to alert business representatives of certain conditions or danger signs (e.g., when used in a message box), or to present a graphical image (e.g., when used in place of a pushbutton or to represent an application). Logos, maps, and photographs also fall into the icon category.

An icon is a type of resource. Resources are bitmapped graphics that are used in windows to represent information pictorially. Usually, the word "icon" refers to a logical icon. Logical icons can be divided into physical bitmaps and physical icons. A physical bitmap is a representation of a picture that can be stored in memory or on a disk. A physical icon is a type of bitmapped graphic.

Common types of icons include application icons, selection cursor icons, pushbutton icons and static-window bitmaps. Windowed environments usually provide standard icons, but a business representative can define custom icons with an icon editor or a graphics tool.

Description

An icon is a bitmapped graphical symbol used to represent some object, function, data structure, or user interface element. When creating an Icon Definition, one set of definitions must be established:

I: A general description of the icon's functionality: In this section (icon window), all the information necessary to describe icon HOTEL is displayed. The widget name (HOTEL), the type of icon being defined (bitmapped), the name of the dynamic link library containing HOTEL (D1ICONS), the resource ID used to identify HOTEL (1), and HOTEL's display size (two fractional character units high by five fractional character units wide) all have been defined. In addition, the height and width of HOTEL are defined to be overridden by the business representative.

[1] Icon ID: Identifier of the icon, as referenced by the computer aided software engineering (CASE) tool

[2] Widget Name: Name used to refer to the icon within a resource file

[3] Type: Type of icon being used (image resource, icon, or bitmap)

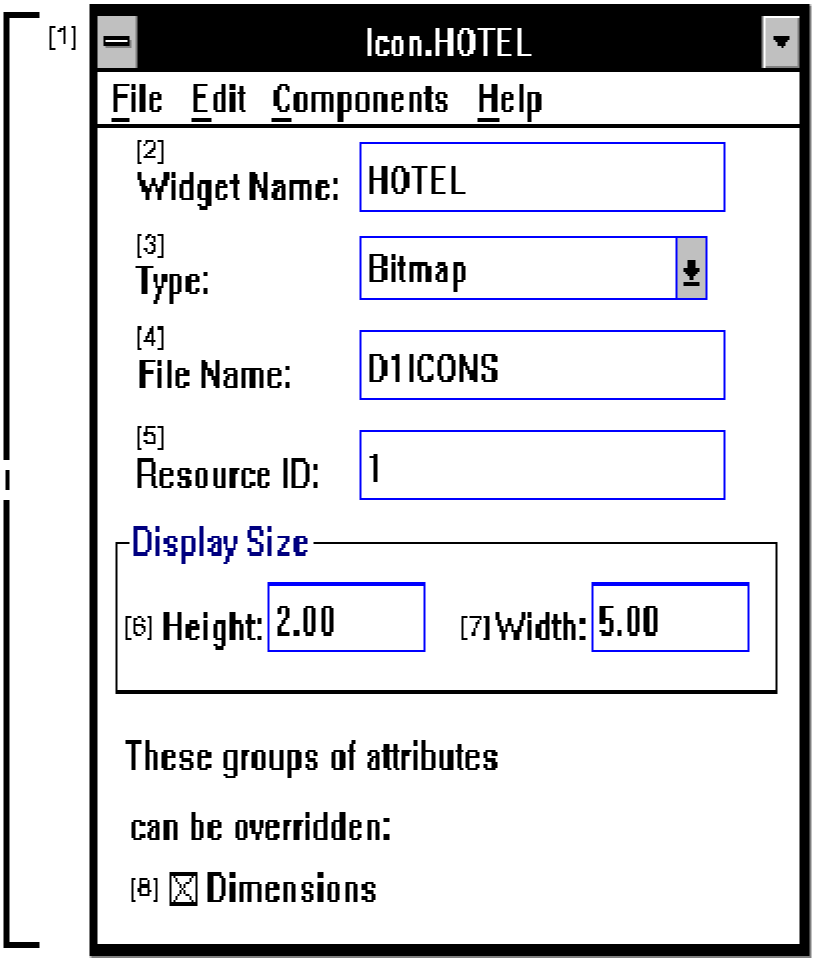
[4] File Name: Name of the resource file in which the icon or bitmap is stored

[5] Resource ID: Unique number by which the operating system can identify the icon in a Dynamic Link Library or an executable that can be compiled

[6] Display Height: Height of the icon's bitmap, measured in fractional character units

[7] Display Width: Width of the icon's bitmap, measured in fractional character units

[8] Override Dimensions: Indicator of whether the icon's display size can be overridden on a window



[2] Widget Name: HOTEL

[3] Type: Bitmap

[4] File Name: D1ICONS

[5] Resource ID: 1

Display Size

[6] Height: 2.00 [7] Width: 5.00

These groups of attributes can be overridden:

[8] ☒ Dimensions

List Box Definition

A List Box Definition specifies a box (usually with scroll bars) that contains a list of choices from which a business representative can select. Use this deliverable to create a list box widget that is used on windows to display multiple rows of data. List boxes can be used for displaying dynamic data (when it is impossible to know ahead of time how many rows will be needed), displaying many rows of static data, and selecting from multiple rows of data. List Box Definitions can also serve as input to a code generator.

Description

[1] List Box Name: Descriptive name of the list box as referenced by the computer aided software engineering (CASE) tool

[2] Title: Text associated with the list box

[3] Column Headings: Indicator of whether headings should appear above the vertical columns on the list box

[4] Horizontal Lines: Indicator of whether horizontal lines should be displayed to separate the list box rows on the window

[5] Vertical Lines: Indicator of whether vertical lines should be displayed to separate the list box columns on the window

[6] Data Element ID: Object name for the data element defining a column

[7] Heading: Title of a list box column

[8] Last Column in Row: Indicator of whether the specified column is the last column in the current row of the list box. If checked, the following column will be placed in the next row's first column.

[9] Hidden Column: Indicator of whether the specified column will be hidden

[10] Non-Scrollable Column: Indicator of whether the specified column is scrollable or always remains static

[11] Display Type: Indicator of whether the specified column will be represented by text or an icon on the screen

[12] Alignment: Field that sets the justification of a cell within a column (left, center, or right)

[13] Sort List Box By This Column: Indicator of whether the list box will be sorted using the specified column as the primary sort key

[14] Order: Order that the data within the list box will be sorted in (ascending or descending). This field is only enabled if [12] has been checked

[15] Width: Width of the list box or column in application character units. Setting this field to zero will cause the width to default to the size of the column heading or the length of the data element, whichever is larger.

[16] Length: Number of characters allotted for the value of a column in the list box

[17] Precision: Number of decimal places allowed for use within the data element

[18] Format: Type of information allowed in this data element (alphanumeric, numeric, graphic, or mixed)

[19] Data Type: Data type of the data element for this column

[20] Structure: Code that describes the way data should be formatted and presented in the list box column

[21] Foreground: Foreground color of the list box column data and heading text

[22] Background: Background color of the list box column interior

[23] Column Font Name: Font name of the text appearing in the list box column

[24] Widget Name: Unique widget name of the list box. This field also acts as the identifier for the list box in a resource file.

[25] C Prefix/Name: C prefix and C name by which the list box can be identified in a C function

[26] COBOL Prefix/Name: COBOL prefix and name by which the list box can be identified in a COBOL program

[27] Height: Height or length of the list box, listed in application character units

[28] Title: Indicator of whether the title for this list box can be overridden on a window in which the list box is painted

[29] Callbacks: Indicator of whether the callbacks for this list box can be overridden on a window in which the list box is painted

[30] Size: Indicator of whether the width and height for this list box can be overridden on a window in which the list box is painted

List boxes have one or more columns and an undetermined number of rows. Selecting an item within the list box selects all corresponding items in the same row as the selection. The selected row can be used for data entry, to control the next step of the dialog, or for other similar purposes.

Several components must be defined when creating a List Box Definition:

I: A description of the list box's overall display attributes: This section (the List Box window) describes the basic display attributes for list box ROOMS AVAILABLE: the literal title of the list box ("Rooms Available"), the display settings for rows and columns (display column headings and vertical column separators), the names of all the data elements that define columns in the list box (# ROOMS, ROOM TYPE, PREFERENCES, and RATE), and the column headings (each column heading is the same as its corresponding data element name).

II: A list of each column's specific display attributes: In this section (the Columns window), the PREFERENCES column within ROOMS AVAILABLE is given more specific display parameters. Display type (normal text), text alignment (center), list box sorting information (PREFERENCES will be the column used as the sort key, and it will be sorted in an ascending order), and column width (10 fractional character units) are all defined here.

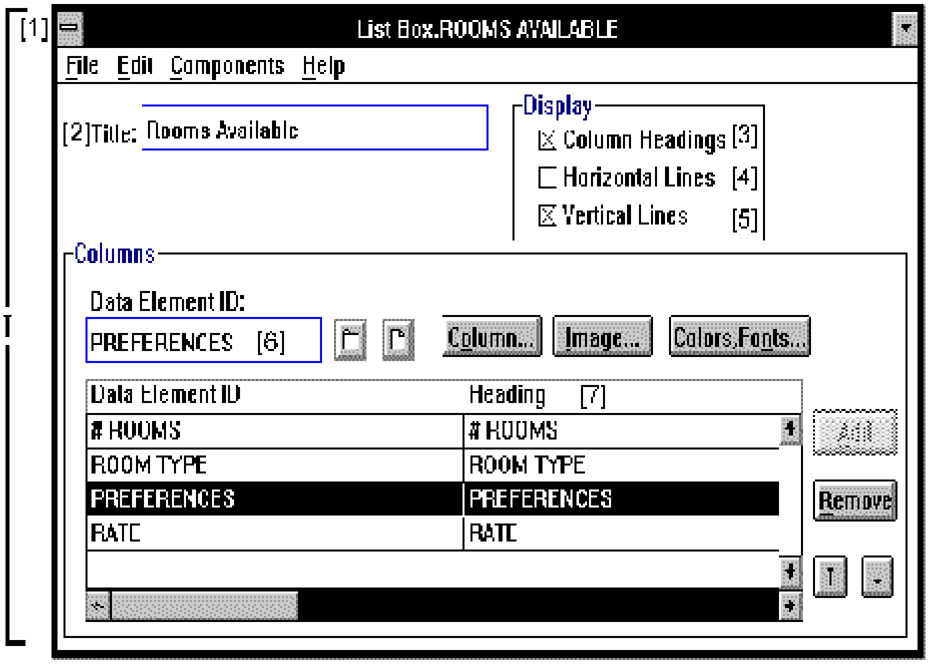
III: A description of each column's image parameters: In this section (the Column Image window), the display parameters of the PREFERENCES column are further defined. The number of characters allowed for display (10), the number of decimal places allowed for a numeric value (0), the format and data type (both alphanumeric), and structure (number 48 in the repository), are all defined for the PREFERENCES column within this section.

IV: A list of the colors used within each column: This section (the Colors and Fonts window) specifies the type of text used within the PREFERENCES column. The text will be 10 point Helvetica Bold Italic in black, with a white background.

V: A list of the generation names used within the list box: This section (the Generation Names window) lists the prefixes and reference names for list box ROOMS AVAILABLE within the C language (RoomsAvailable) and the COBOL language (ROOMS-AVAILABLE). The internal widget name for the list box also is defined here (ROOMS_AVAILABLE).

VI: The size of the list box: In this section (the Size window), the height and width of ROOMS AVAILABLE are determined: 80 fractional character units wide by 10 fractional character units tall.

VII: A list of the attributes that can be overridden within the list box: This section (the Overrides window) shows that the title, callbacks, and size of ROOMS AVAILABLE can all be overridden by the business representative when the list box is painted on a window or screen.

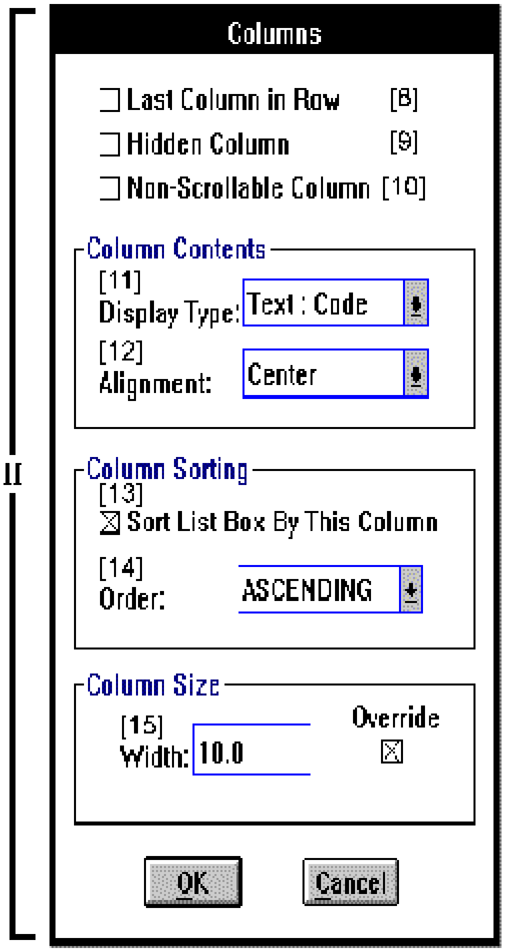


[2]Title: Rooms Available

☒ Column Headings [3]
□ Horizontal Lines [4]
☒ Vertical Lines [5]

Data Element ID: PREFERENCES [6]

Data Element ID	Heading [7]
# ROOMS	# ROOMS
ROOM TYPE	ROOM TYPE
PREFERENCES	PREFERENCES
RATE	RATE



Columns

□ Last Column in Row [8]

□ Hidden Column [9]

□ Non-Scrollable Column [10]

Column Contents

[11] Display Type: Text : Code

[12] Alignment: Center

Column Sorting

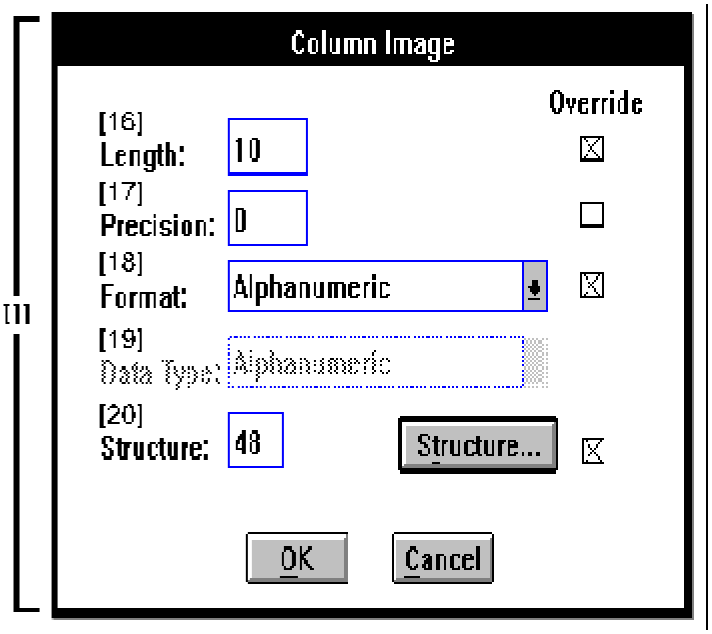
[13] ☒ Sort List Box By This Column

[14] Order: ASCENDING

Column Size

[15] Width: 10.0

Override ☒



Column Image

Override

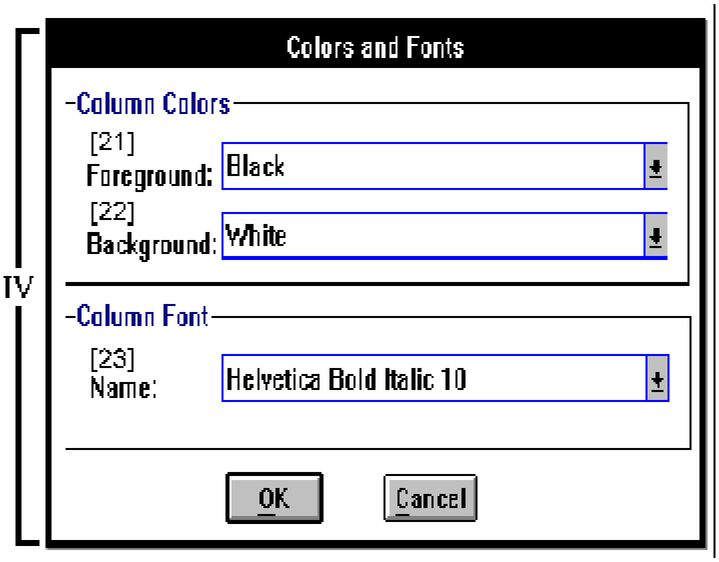
[16] Length: 10 ☒

[17] Precision: 0 □

[18] Format: Alphanumeric ☒

[19] Data Type: Alphanumeric

[20] Structure: 48 ☒



Colors and Fonts

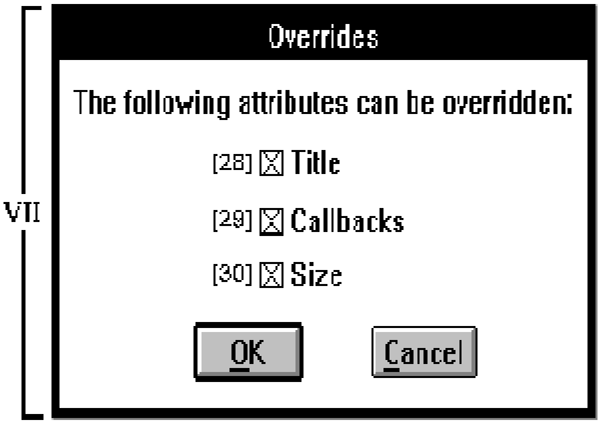
Column Colors

[21] Foreground: Black

[22] Background: White

Column Font

[23] Name: Helvetica Bold Italic 10



Overrides

The following attributes can be overridden:

[28] ☒ Title

[29] ☒ Callbacks

[30] ☒ Size

Menu Item Definition

A Menu Item Definition describes a command or other element on a menu that, when selected, the menu item leads directly to a specific action. This deliverable is a component of the Widget Definitions composite deliverable. Use this deliverable to define the appearance, contents, and functions of a menu item. A Menu Item Definition can define items such as separators between menu items, submenus within the main menu, events and callbacks for the menu and its submenus, and the appearance of the literals. This description also can serve as input to a code generator.

A menu displays a list of options available for a selected choice in the action bar of a window. A menu can selectively provide additional options within a submenu. Cascading menus are pulldown menus that are invoked from a submenu item within a pulldown menu. In many applications, a cascading menu is displayed when a business representative selects a menu item with a right arrow displayed to its right.

Keyboard shortcuts, such as Alt+S for Save, can be defined for any menu item. They allow business representatives to keystroke rather than use the mouse. Providing timesaving shortcuts is particularly important when the application will be used by experienced business representatives.

Typically, the Menu Item Definition is created by the user interface designer.

Description

[1] Menu Item ID: Identifier of the menu item as referenced by the computer aided software engineering (CASE) tool

[2] Widget Name: Unique widget name of the menu item, as it appears in a resource file

[3] Literal: Text that will appear on the menu to identify the item

[4] Alt, Ctrl, Shift: These fields are used to help identify a keyboard shortcut that may be used to invoke the menu item. Each marked field indicates that the corresponding key (Alt, Ctrl, and/or Shift) must be held down before pressing the key indicated in [5] to invoke the shortcut.

[5] Key: Key used to trigger the keyboard shortcut. Note that when defining the shortcut, any key used to enter text (A-Z, 0-9, and punctuation keys) must be coupled with at least one of the special keys listed in [4]. ENTER, ESC and function keys also may be specified as valid shortcut keys.

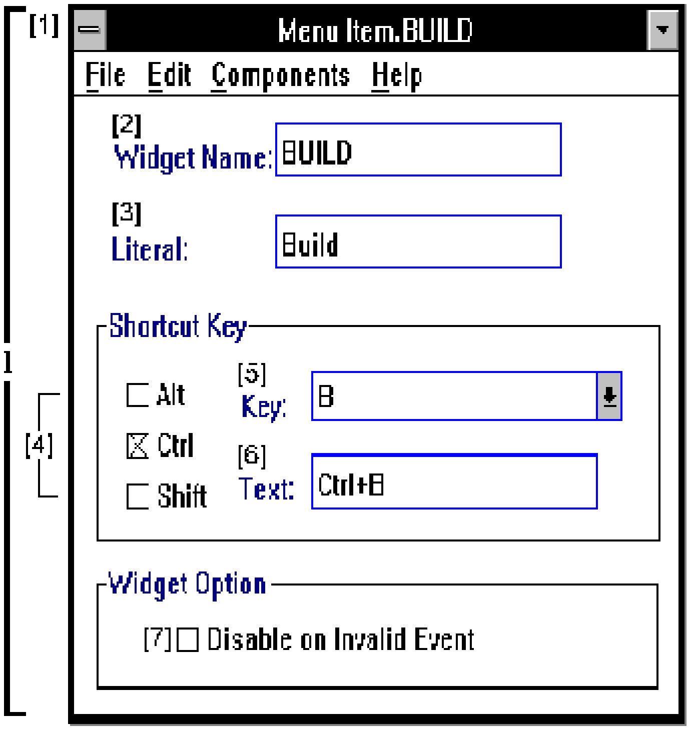
[6] Text: Text appearing to the right of the menu item, identifying the item's keyboard shortcut. This value defaults to your combined key selection.

[7] Disable on Invalid Event: Indicator of whether the menu item will be automatically disabled when a widget triggers an invalid event

[8] Override Callbacks: Indicator of whether menu item callbacks can be overridden
When creating a Menu Item Definition, the following definitions should be included:

I: A general description of the menu item's functionality: This section (the Menu Item window) provides a basic description of menu item BUILD. BUILD's literal ("Build") and the shortcut key used to invoke BUILD from the keyboard (ctrl+B) are defined here, as are the name of the widget triggered by BUILD (the BUILD widget), and disabling instructions for the menu item (BUILD will be disabled if activating the BUILD widget would trigger an invalid event).

II: A list of all overridable attributes of the menu item: In this section (the Allow Overrides window), the callbacks invoked by menu item BUILD, i.e., the code triggered when the item is selected, are defined to be overridable by the business representative.



[2] Widget Name: BUILD

[3] Literal: Build

Shortcut Key

□ Alt

☒ Ctrl

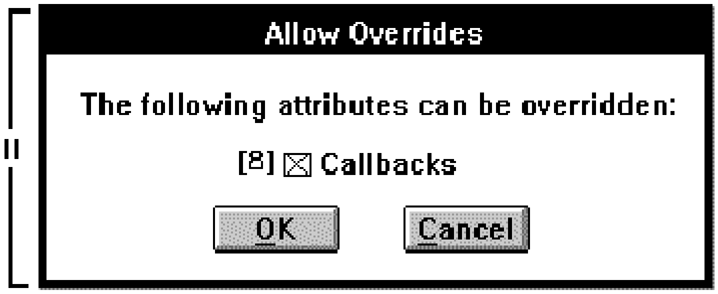
□ Shift

[5] Key: B

[6] Text: Ctrl+B

Widget Option

[7] □ Disable on Invalid Event



Allow Overrides

The following attributes can be overridden:

[8] ☒ Callbacks

Multi-Line Entry Field Definition

A Multi-Line Entry Field Definition describes an input/output (I/O) data element displayed as a multi-line entry field widget. Use this deliverable in graphical user interfaces for multi-line entry field I/O data elements.

In client/server applications, most data elements will appear on one or more windows in the application. Depending on the nature of the data element, it is represented by a particular type of widget: single-line entry field, multi-line entry field, radio button group, combo box, check box, or slider. The definitions associated with the multi-line entry field widget type are listed below.

Description

XI: A list of display attributes for the data element's entry field: In this Multi-Line Entry Field Definition sample, a data element designed to hold a standard introductory paragraph for letters to customers has been partially defined. The display attributes of this multi-line entry field data element have been established: The number of characters allowed on a line (5), the width and height of the entry field in character units (20 and 10, respectively), and the number of spaces inserted when the Tab key is pressed (5). Note that this specifies only the display size of the field. The number of characters that may be used in defining the data element's value is defined in another section.

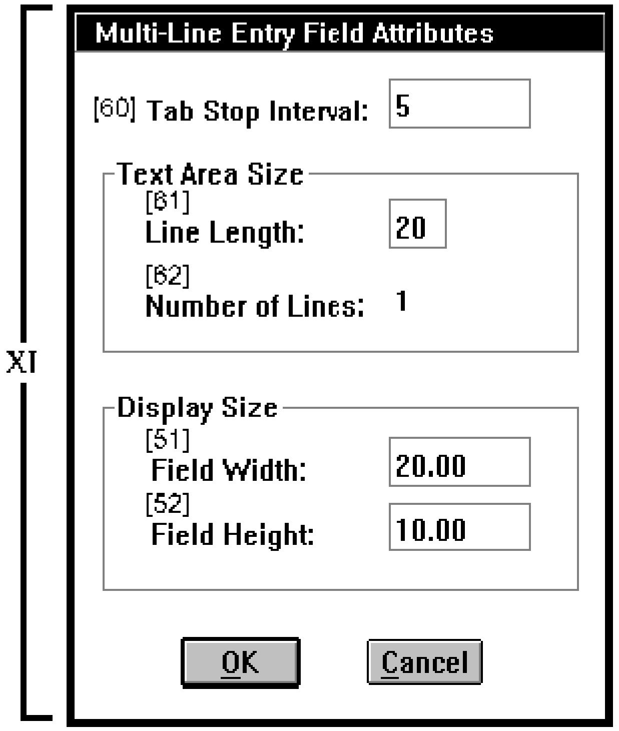
[51] Field Width: Default width of a single-line entry field, measured in character units. Note that this refers only to the display size of the entry field, and has no bearing on the number of characters the field can actually contain.

[52] Field Height: Default height of a single-line entry field, measured in character units. Note that this only refers to the display size of the entry field, and has no bearing on the number of characters the field can actually contain.

[60] Tab Stop Interval: Number of spaces the cursor will advance when the Tab key is pressed

[61] Line Length: Maximum number of characters allowed on a text line before a carriage return is needed, or before word wrap automatically inserts a carriage return.

[62] Number of Lines: Total number of lines allowed within a multi-line entry field, calculated by dividing screen/window length by line length. This field is for display purposes only and cannot be directly manipulated by the business representative.



Multi-Line Entry Field Attributes

[60] Tab Stop Interval: 5

Text Area Size

[61] Line Length: 20

[62] Number of Lines: 1

Display Size

[51] Field Width: 20.00

[52] Field Height: 10.00

Pushbutton Definition

A Pushbutton Definition describes an object used on a window to invoke an event that can manipulate or validate data on a window, activate another window, or call a server program. When designing a graphical user interface, use a Pushbutton Definition to define the components of the pushbutton widget. This description also can serve as input to a code generator.

Pushbuttons are used to invoke a command or option. They are particularly useful for providing access to actions such as OK, Cancel, Apply, and Help. Windows often contain one or more pushbuttons. When clicked, a pushbutton causes something to happen immediately (e.g., a window is closed or opened). In this respect, it is similar to a function key.

Shortcut keys, such as Ctrl+O for OK, can be defined for pushbuttons. They allow business representatives to keystroke rather than use the mouse. Providing timesaving shortcuts is particularly important when the application will be used by experienced business representatives.

Some tools allow defaults to be assigned for the Enter and Escape keys. The defaults are the pushbuttons that will be activated when the business representative presses Enter or Esc. Only one default pushbutton of each type should be allowed. Defaults specified here can be overridden by the associated Window Description.

Typically, the Pushbutton Definition is created by the user interface designer.

Description

[1] Pushbutton ID: Identifier of the pushbutton as referenced by the computer aided software engineering (CASE) tool

[2] Widget Name: Unique widget name of the pushbutton as it appears in a resource file

[3] Literal: Literal that will appear on the pushbutton when it is displayed on a window

[4] Alt, Ctrl, Shift: Fields used to help identify a keyboard shortcut that may be used to invoke the pushbutton. Each marked field indicates that its corresponding key (Alt, Ctrl, and/or Shift) should be held down before pressing the key indicated in [5] to invoke the shortcut.

[5] Key: Key used to trigger the keyboard shortcut. Note that any key used to enter text (A-Z, 0-9, and punctuation keys) must be coupled with at least one of the special keys listed in [4] when defining the shortcut. ENTER, ESC, and the function keys also may be specified as valid shortcut keys.

[6] Disable on Invalid Event: This field determines whether the pushbutton will be disabled when an invalid event is triggered by a widget on the window.

[7] Width: Width of the pushbutton in character units

[8] Height: Height of the pushbutton in character units

[9] Iconic: This field determines whether the pushbutton will be displayed with an icon, rather than with its literal

[10] Icon: Name of the icon to be displayed in place of the pushbutton literal. This field is active only when [9] is checked.

[11] Callbacks: Indicates whether the defined pushbutton callback can be overridden within a window

[12] Literal: Indicates whether the defined pushbutton literal can be overridden within a window

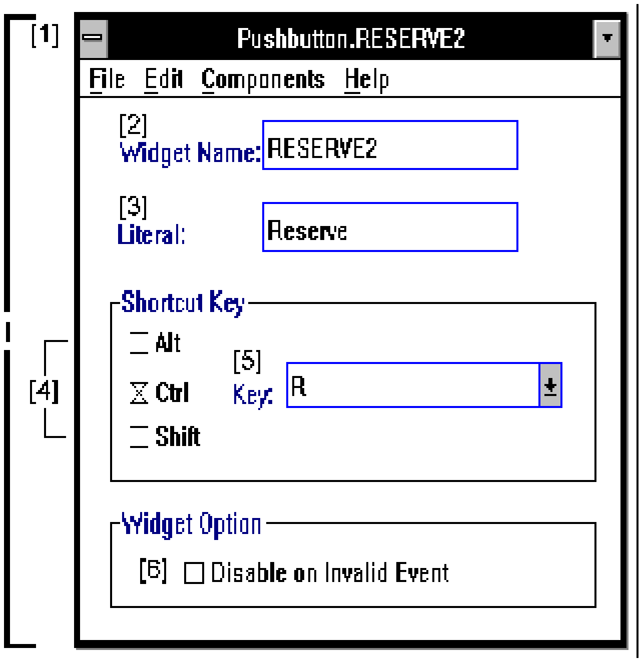
[13] Size: Indicates whether the defined pushbutton height and width can be overridden within a window

When creating a Pushbutton Definition, include the following components:

- **I: A general description of the pushbutton's functionality:** This section (the Pushbutton window) lists the most basic information about pushbutton RESERVE2: its widget name (RESERVE2), the literal appearing on the button itself ("Reserve"), the shortcut key used to invoke the button from the keyboard (ctrl+r), and disabling instructions (the pushbutton will not be disabled in cases where it would activate an invalid event).

- **II: A list of the pushbutton's display attributes:** In this section (the Display Attributes window), the height and width of pushbutton RESERVE2 are defined (12.5 fractional character units wide; the height will default to the minimum size needed to fit the literal on the pushbutton). If RESERVE2's literal were to be replaced with an icon, the repository name for the icon would also be listed here.

- **III: A list of any of the pushbutton's attributes that can be overridden by the developer:** This section (the Allow Overrides window) shows that the pushbutton's callbacks, literal, and size can all be overridden by the developer when RESERVE2 is painted on a window.



[2] Widget Name: RESERVE2

[3] Literal: Reserve

Shortcut Key

□ Alt

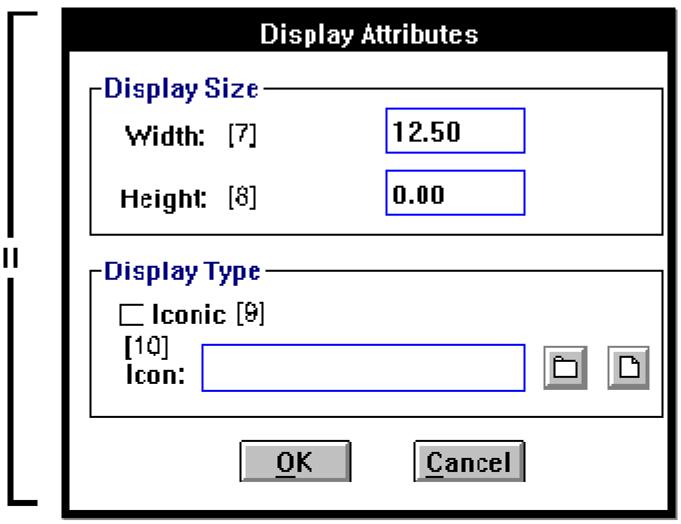
☒ Ctrl

□ Shift

[5] Key: R

Widget Option

[6] □ Disable on Invalid Event



Display Attributes

Display Size

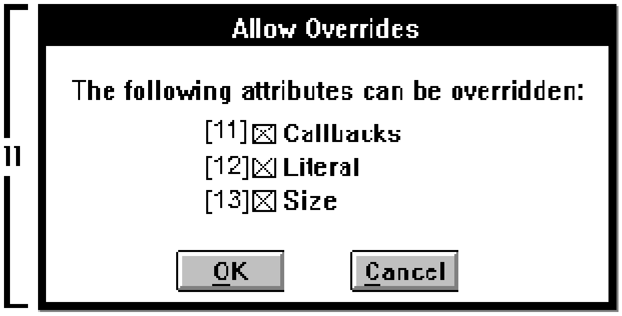
Width: [7] 12.50

Height: [8] 0.00

Display Type

□ Iconic [9]

[10] Icon:



Allow Overrides

The following attributes can be overridden:

[11] ☒ Callbacks
[12] ☒ Literal
[13] ☒ Size

Radio Button Definition

A Radio Button Definition describes an input/output (I/O) data element displayed as a radio button widget. Use this deliverable in graphical user interfaces for radio button I/O data elements

Description

In client/server applications, most data elements will appear on one or more windows in the application. Depending on the nature of the data element, it is represented by a particular type of widget: single-line entry field, multi-line entry field, radio button group, combo box, check box, or slider. The definitions associated with the radio button widget type are listed below.

XII: A list of the values that can be selected within a radio button group: In this sample, the values that can be selected for a radio button data element (designed to offer the business representative a choice of bed sizes in a hotel) have been defined. Each member of the radio button group is identified in two terms: the literal that will appear next to the button, and the value sent to the data element if the button is selected. For example, if the business representative selects the Queen button, the data element will contain a value of Queen. Language-specific references for each button value also may be defined within this section. For example, in COBOL, the reference for the Queen radio button is defined as BED-SIZE-QUEEN.

XIII: A list of the icons, if any, that will replace the literals of specified radio buttons: This section (the Valid Val window) shows that within the radio button group specified within data element NAME2, icon SINGLE-BED-ICON will be displayed next to the Single button, in place of the literal "Single."

[2] Literal: The literal, on a screen or a report, identifying the data element.

[13] COBOL Name: COBOL name of the 88-level item

[20] Widget Name: Name used when the element is generated and programmatically manipulated as an item on a window

[22] COBOL Name: Name used to access the data element within the COBOL programming language

[45] Value: An individual value within a list of values that describes the domain of values that are valid for the data element.

[46] Use as Default Active: Indication that the value is part of the subset of values that represents the default validation rules for the data element. This is a subset of values that should be used in 99% of the cases for the element. If this indicator is not set, then it is a value that is valid for the element, but is only valid in extreme cases.

[47] Validation: Indicator of whether the value in [45] is part of a list of values against which the data element should be validated. If this indicator is not set, then this value is one value that the data element may take on, but the element is not limited to these values. This field is used to indicate initial values for data elements that may have no limiting set of validation rules.

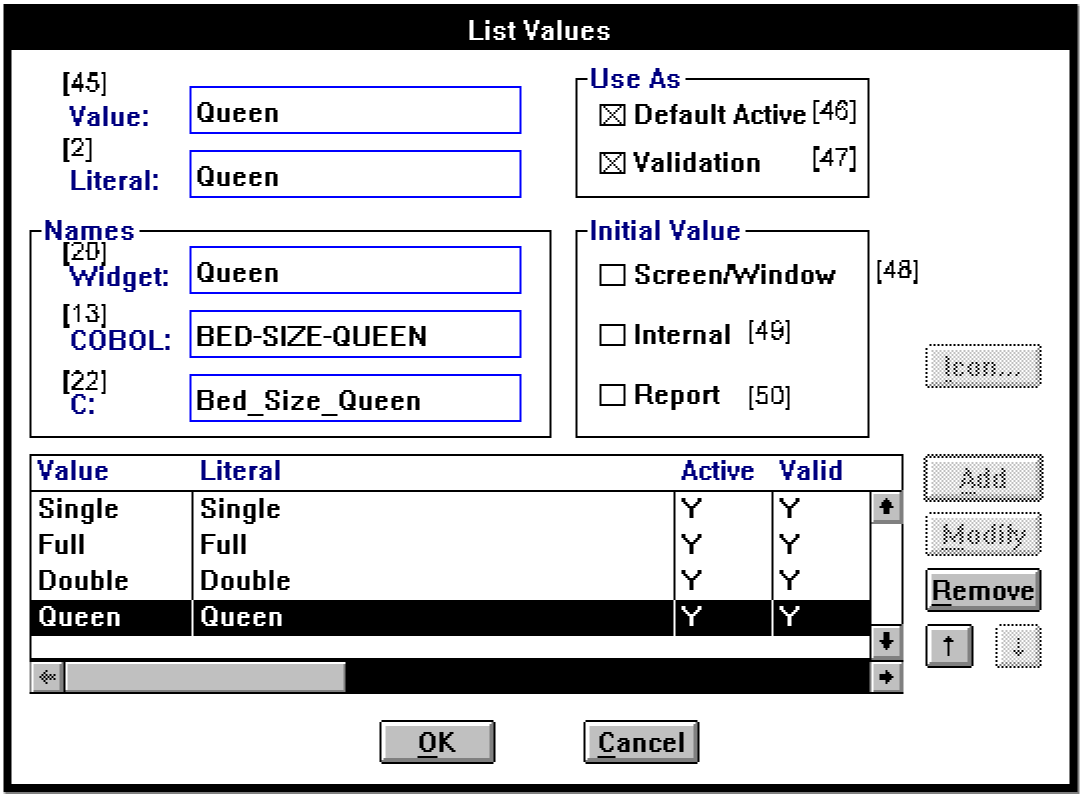
[48] Initial Screen/Window Value: Indicator of whether the data element should be set to the value in [45] when the data element is initially displayed on a screen or window

[49] Initial Internal Value: Indicator of whether the data element should be initialized to the value in [45] when the field is defined in internal memory or in a file or relational table

[50] Initial Report Value: Indicator of whether the data element should be set to the value in [45] when the data element is initially displayed on a report

[63] Icon: Name of an icon to be associated with a list value of the data element

XII



List Values

[45] Value: Queen

[2] Literal: Queen

Use As

☒ Default Active [46]

☒ Validation [47]

Names

[20] Widget: Queen

[13] COBOL: BED-SIZE-QUEEN

[22] C: Bed_Size_Queen

Initial Value

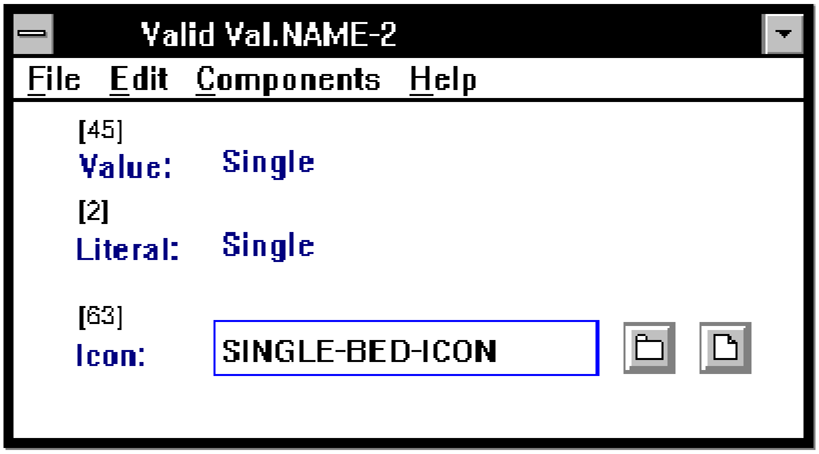
□ Screen/Window [48]

□ Internal [49]

□ Report [50]

Value	Literal	Active	Valid
Single	Single	Y	Y
Full	Full	Y	Y
Double	Double	Y	Y
Queen	Queen	Y	Y

XIII



Valid Val.NAME-2

[45] Value: Single

[2] Literal: Single

[63] Icon: SINGLE-BED-ICON

Single-Line Entry Field Definition

A Single-Line Entry Field Definition describes an input/output (I/O) data element displayed as a single-line entry field widget. Use this deliverable in graphical user interfaces for single-line entry field I/O data elements.

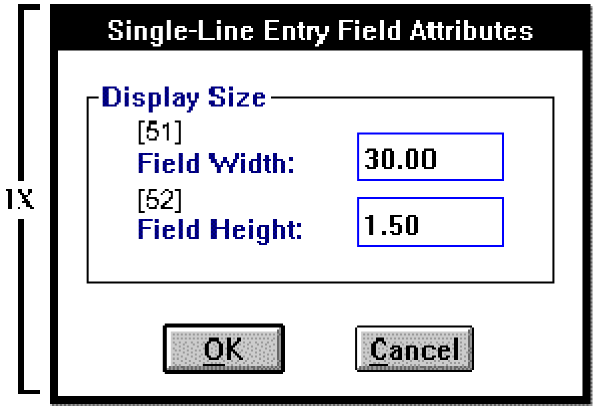
Description

In client/server applications, most data elements will appear on one or more windows in the application. Depending on the nature of the data element, it is represented by a particular type of widget: single-line entry field, a multi-line entry field, radio button group, combo box, check box, or slider. The definitions associated with the single-line entry field widget type are listed below.

IX: A list of display attributes for the data element's entry field: This sample defines the height and width of CUSTOMER AGE's entry field: 30 character units wide and 1.5 character units tall. Note that this specifies only the display size of the field. The number of characters that may be used in defining the data element's value is defined in another section.

[51] Field Width: Default width of a single-line entry field, measured in character units. Note that this refers only to the display size of the entry field, and has no bearing on the number of characters the field can actually contain.

[52] Field Height: Default height of a single-line entry field, measured in character units. Note that this refers only to the display size of the entry field, and has no bearing on the number of characters the field can actually contain.



Single-Line Entry Field Attributes

Display Size

[51]
Field Width: 30.00

[52]
Field Height: 1.50

Slider Definition

A Slider Definition describes an input/output (I/O) data element displayed as a slider widget. Use this deliverable in graphical interfaces for slider I/O data elements.

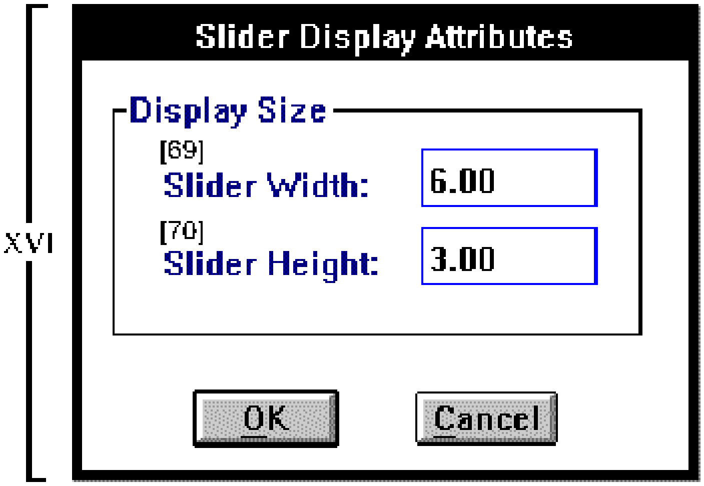
Description

In client/server applications, most data elements will appear on one or more windows in the application. Depending on the nature of the data element, it is represented by a particular type of widget: single-line entry field, multi-line entry field, radio button group, combo box, check box, or slider. The definitions associated with the slider widget type are listed below.

XVI: A description of the display attributes for the slider: This sample defines the height and width of a slider used to set the preferred temperature in a hotel guest's room. In the example, the height has been set to three character units, while the width has been set to six character units.

[69] Slider Width: Width of a slider as it appears on a screen or window, measured in character units

[70] Slider Height: Height of a slider as it appears on a screen or window, measured in character units


Slider Display Attributes

Display Size

[69] Slider Width: 6.00

[70] Slider Height: 3.00

Window Definition

A Window Definition graphically represents what the window will look like when painted on the screen. It documents the layout and behavior of a window, and highlights window development standards. Use this deliverable to record the design for new windows within an application or changes to existing windows in a packaged software application (Some packaged software applications may refer to a window definition as a screen or a form). If a code generator is being used, coordinate with its requirements so that the window can serve as an input to the generator.

Description

The **General** area of the sample documents configuration management and audit trail information concerning this document. The **Summary** area of the sample documents the name, a short paragraph, and a list of keywords that describe the content of this document.

Note: When using the BI Designer Toolset, the fields in these areas are automatically provided.

Link to CAR Diagram: Reference to a deliverable document describing the widgets on the window, the possible actions on the widgets and the response to those actions

Data Sources: Source of data for a widget

> **Widget Name:** Name of the widget
>
> **Widget Type:** The type of widget. Typical values are Window, Menu Item, Push Button, List Box, Entry Field, Check Box, Radio Button, and Slider.
>
> **Data Element:** Name of or reference to a deliverable document describing the data element that supplies data to this widget
>
> **Table:** Name of or reference to a deliverable document describing the source of data for this widget
>
> **Type of Table:** Type of object identified in Source Table/File Name
>
> **Tab Order:** Sequence order of the tabbing on the widgets on the window
>
> **Search field?:** Indicates whether the data element is part of the search key

Window Layout: Design of what the window looks like

Performance Considerations: Length of time required to create this window and the effect of creating the window on other aspects of the application

Security Considerations: Security issues for those who can create this window

Error Handling: Description of potential errors within the window and the determination if specific notification procedures are required

Assumptions: Documentation of any assumptions for which the window design is based

Business Logic/Calculation Rules: Logic for all fields being calculated by the extract including totaling logic and summary levels

An operating system can provide multiple windows on a single screen, allowing a business representative to keep several applications active and visible simultaneously. Individual applications can also provide multiple windows, providing viewing capability to more than one document, spreadsheet, or file. Each window contains a varying number of predefined widgets. When creating a window layout, establish the following information:

- **Window Layout and Contents:** In the sample, window RECONNECT ACCOUNT INFORMATION has been visually represented. The placement of entry fields (including Billing Address and Alternate Name), radio button groups (Revenue Class and Deposit Method), pushbuttons (OK and Cancel), and a list box (Required Task) have been positioned here.

- **Window Characteristics:** This information should include the window's style (top, child, dialog, etc.), modality (determines whether the window can remain displayed while the business representative operates on other windows in the application), border attributes (size and resizability), and manipulation characteristics (scroll bars, minimize/maximize boxes, entry field cursors, alignment to other windows, etc.).

Window Definition

General

Type:	Window Definition
Version number:	1.0
Version labels:	1.0 CURRENT

Created:	08/03/98 04:03 PM
Modified:	10/15/98 11:55 AM
Last modified by:	UserX
Created by:	UserX

Summary

Name:	Reconnect Account Information
Title (Description):	The Reconnect Account Information window is used to reestablish service for a disconnected customer

Keywords:

Details

Link to CAR Diagram

Name:	Reconnect Account Information

Data Sources

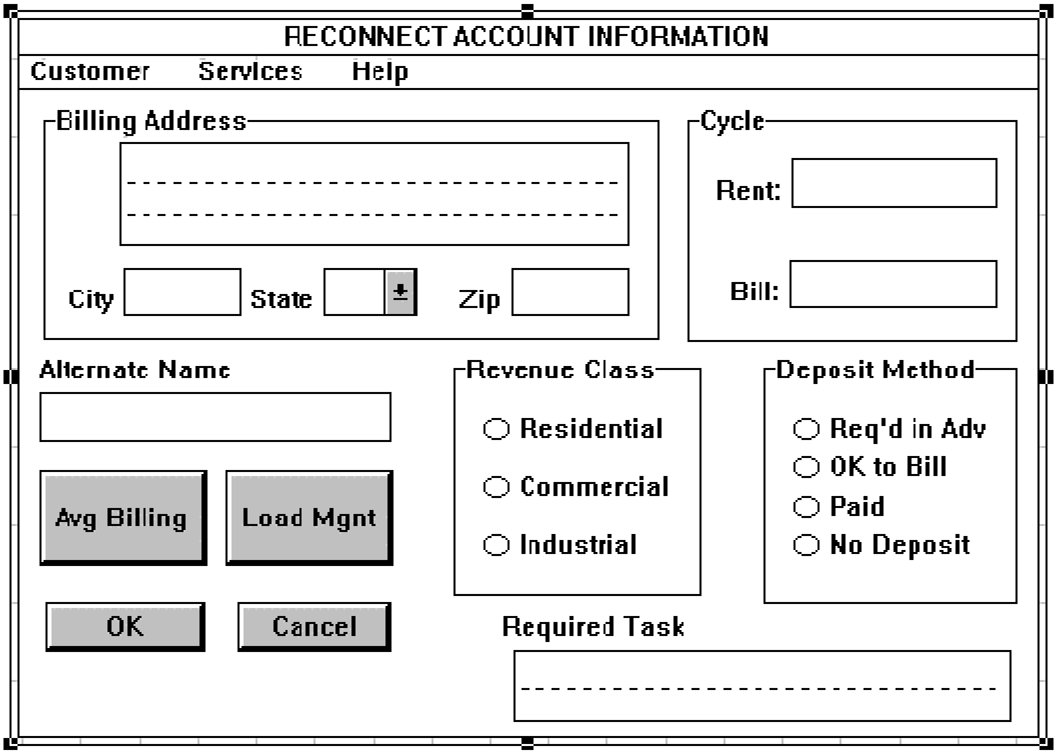
List of data sources:

Widget name	Widget type	Data element	Table	Type of Table	Tab order	Search field?
Address	Entry Field	Address	Acct Info	File	1	
City	Entry Field	City	Acct Info	File	2	
State	Combo Box	State	Acct Info	File	3	
Zip	Entry Field	Zip	Acct Info	File	4	
Rent	Entry Field	Rent	Acct Info	File	5	
Bill	Entry Field	Bill	Acct Info	File	6	
Alternate Name	Entry Field	Bill To Name	Acct Info	File	7	Yes
Revenue Class	Radio Button Group	Account Class	Acct Info	File	8	
Deposit Method	Radio Button Group	Deposit Type	Acct Info	File	9	
Required Tasks	Multiple-Line Entry Field	Required Tasks	Acct Info	File	10	
Avg Billing	Push Button	Avg Billing	Acct Info	File	11	

Additional Information

The following section can be used to provide additional information. It is free text only and will not be stored in the associated property pages.

Window Layout



Performance Considerations

Window takes two seconds to load; no adverse effect on application processing anticipated.

Security Considerations

Error Handling

Require entry of billing address; compare billing address with service address for validation.

Assumptions

Billing address will always be supplied by the customer.

Business Logic/Calculation Rules